

BEST AVAILABLE COPY

[Extract Translation of Japanese Patent Publication No.H11-317783]

[Title] A header processing apparatus and a method thereof

[Abstract]

[Object] A pipeline processing method is provided. The pipeline processing method uses a plurality of general CPUs or a sequencer for the rapid processing of a packet header .

[Construction] A header processing apparatus and method comprises: first means 101 for extracting a field data of a protocol header from a part of a packet including a header divided from the packet, and inputting the extracted field data to an assigned memory; second means 102 for detecting data from the assigned memory using a detection key, and outputting a content of a table entry corresponding to the detection key; and third means 103 for preparing attribute information about a destination of the packet and about header update information, based on the an output from the second means and the data of the memory.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-317783

(43) 公開日 平成11年(1999)11月16日

(51) Int.Cl. ⁶	識別記号	F I		
H 0 4 L	29/02	H 0 4 L	13/00	3 0 1 Z
	12/28		11/00	3 1 0 D
	12/56		11/20	1 0 2 A

審査請求 有 請求項の数17 O L (全 18 頁)

(21) 出願番号 特願平11-16006
 (22) 出願日 平成11年(1999)1月25日
 (31) 優先権主張番号 特願平10-11437
 (32) 優先日 平10(1998)1月23日
 (33) 優先権主張国 日本 (J P)

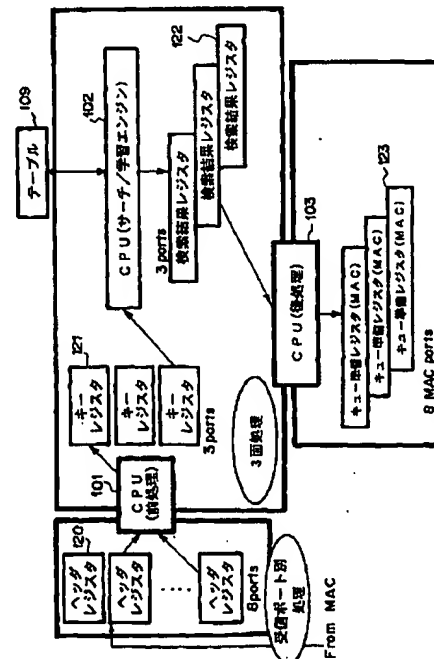
(71) 出願人 000004237
 日本電気株式会社
 東京都港区芝五丁目7番1号
 (72) 発明者 村瀬 勉
 東京都港区芝五丁目7番1号 日本電気株式会社内
 (72) 発明者 小林 正好
 東京都港区芝五丁目7番1号 日本電気株式会社内
 (74) 代理人 弁理士 山下 穰平

(54) 【発明の名称】 ヘッダ処理装置とそのヘッダ処理方法

(57) 【要約】

【課題】 複数の汎用CPUやシーケンサを用い、パケットヘッダ処理の高速化のためパイプライン処理等の方法を提供する。

【解決手段】 ヘッダ処理装置及び該方法において、パケットから抽出されたヘッダを含むパケットの一部分から、プロトコルヘッダの各フィールドのデータを抽出し、指定された記憶場所に入力する第1の手段101と、該指定された記憶場所のデータを検索キーにして検索を行い、検索キーに一致するテーブルエントリの内容を出力する第2の手段102と、該第2の手段の出力内容と前記記憶場所のデータとから、前記パケットの送出先とヘッダ更新情報の属性情報を作成する第3の手段103と、を有することを特徴とする。



【特許請求の範囲】

【請求項1】 パケットから抽出されたヘッダを含むパケットの一部分から、プロトコルヘッダの各フィールドのデータを抽出し、指定された記憶場所に入力する第1の手段と、

該指定された記憶場所のデータを検索キーにして検索を行い、該検索キーに一致するテーブルエントリの内容を出力する第2の手段と、

該第2の手段の出力内容と前記記憶場所のデータとから、前記パケットの送出先とヘッダ更新情報の属性情報を作成する第3の手段と、を有することを特徴とするヘッダ処理装置。

【請求項2】 請求項1に記載のヘッダ処理装置において、前記第1～第3の手段は、1つのシーケンサと2つのCPU、2つのシーケンサと1つのCPU、3つのシーケンサ、3つのCPU、のいずれかにより構成されることを特徴とするヘッダ処理装置。

【請求項3】 請求項1に記載のヘッダ処理装置において、前記第1～第3の手段は、共有されるシーケンサと一つのシーケンサあるいは一つのCPU、共有されるCPUと一つのシーケンサあるいは一つのCPU、のいずれかにより構成されることを特徴とするヘッダ処理装置。

【請求項4】 請求項1～3のいずれか1項に記載のヘッダ処理装置において、前記第1～第3の手段をそれぞれパイプライン処理する手段を有することを特徴とするヘッダ処理装置。

【請求項5】 請求項4に記載のヘッダ処理装置において、前記パイプライン処理について前記第1～第3の手段のいずれかで制御の受け渡しをハンドシェイクを用いて行うことを特徴とするヘッダ処理装置。

【請求項6】 請求項1～5のいずれか1項に記載のヘッダ処理装置において、前記第1～第3の各手段は、それぞれあらかじめ決められた固定時間の始まりあるいは終わりに、現固定時間内に、あるいは次の固定時間内にどの処理プログラムを実行するかを、決定することを特徴とするヘッダ処理装置。

【請求項7】 請求項1～5のいずれか1項に記載のヘッダ処理装置において、前記第1～第3の各手段は、それぞれあらかじめ決められ命令数の整数倍で処理を終了し、処理終了時あるいは開始時に、他の手段の処理の終了あるいは開始を確認し、少なくとも他の1つの手段において終了あるいは開始が確認できない場合には、命令の実行を一時停止し、全手段が処理の終了あるいは開始状態になるのを待つことを特徴とするヘッダ処理装置。

【請求項8】 請求項1～5のいずれか1項に記載のヘッダ処理装置において、前記第1の手段における記憶場所として、1つの共用部分と2つの個別部分を有し、該共用部分は、一部あるいは全てのフィールドデータを前記第2の手段と前記第3の手段で共用して使用するよう

な記憶場所として用い、前記個別部分は、各個別部分に同一内容のデータを持ち且つ各個別部分のそれぞれを、前記第2の手段と、前記第3の手段で独立に用いることを特徴とするヘッダ処理装置。

【請求項9】 請求項2～5のいずれか1項に記載のヘッダ処理装置において、前記第1の手段におけるシーケンサの命令として、ヘッダ可能性部分から、DIX規格とIEEE規格802.3のMAC(Media Access Control)アドレスと、802.1Qと、IPと、TCPと、UDPと等の固定プロトコルフィールドを抽出する特殊演算からなる第1の命令セットと、任意の複数ビットを抽出する基本演算からなる第2の命令セットとを持ち、前記第1の命令セットと前記第2の命令セットの命令を各々固有のハードウェアで処理することを特徴とするヘッダ処理装置。

【請求項10】 請求項2～5のいずれか1項に記載のヘッダ処理装置において、前記第2の手段におけるシーケンサの命令として、テーブルへの検索データ入力と前記テーブルからの検索結果のレジスタへの蓄積を行う特殊演算からなる第1の命令セットと、任意の複数ビットをレジスタからレジスタへ移動する基本演算からなる第2の命令セットとを持ち、前記第1の命令セットと前記第2の命令セットの命令を各々固有のハードウェアで処理することを特徴とするヘッダ処理装置。

【請求項11】 請求項2～5のいずれか1項に記載のヘッダ処理装置において、前記第3の手段におけるシーケンサの命令として、検索結果出力から装置内パケットヘッダの作成を行なう特殊演算からなる第1の命令セットと、任意の複数ビットをレジスタからレジスタへ移動する基本演算からなる第2の命令セットとを持ち、前記第1の命令セットと前記第2の命令セットの命令を各々固有のハードウェアで処理することを特徴とするヘッダ処理装置。

【請求項12】 パケットから抽出されたヘッダを含むパケットの一部分から、プロトコルヘッダの各フィールドのデータを抽出し、指定された記憶場所に入力する第1の手段と、該指定された記憶場所のデータを検索キーにして検索を行い、前記検索キーに一致するテーブルエントリの内容を出力する第2の手段と、該第2の手段の出力内容と前記記憶場所のデータとから、前記パケットの送出先とヘッダ更新情報の属性情報を作成する第3の手段と、を有することを特徴とするヘッダ処理方法。

【請求項13】 請求項12に記載のヘッダ処理方法において、前記第1～第3の手段は、1つのシーケンサと2つのCPU、2つのシーケンサと1つのCPU、3つのシーケンサ、3つのCPU、のいずれかを用いて実行されることを特徴とするヘッダ処理方法。

【請求項14】 請求項12に記載のヘッダ処理方法において、前記第1～第3の手段は、共有されるシーケンサと一つのシーケンサあるいは一つのCPU、共有され

3

るCPUと一つのシーケンサあるいは一つのCPU、のいずれかを用いて実行されることを特徴とするヘッダ処理方法。

【請求項15】 請求項12～14のいずれか1項に記載のヘッダ処理方法において、前記第1～第3の手順をそれぞれパイプライン処理によって行うことを特徴とするヘッダ処理方法。

【請求項16】 請求項15に記載のヘッダ処理方法において、前記パイプライン処理について前記第1～第3の手順のいずれかで制御の受け渡しをハンドシェイクを用いて行うことを特徴とするヘッダ処理方法。

【請求項17】 請求項12～16のいずれか1項に記載のヘッダ処理方法において、前記第1～第3の各手順は、それぞれあらかじめ決められた固定時間の始まりあるいは終わりに、現固定時間内に、あるいは次の固定時間内にどの処理プログラムを実行するかを、決定することを特徴とするヘッダ処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、パケットのヘッダ処理装置及びヘッダ処理方法に関し、特にパケット転送用手段を有してパケットを高速に処理するヘッダ処理装置及びヘッダ処理方法に関する。

【0002】

【従来の技術】従来、公衆電話回線はアナログの音声通信であったが、情報化時代に入ってからデータの通信が盛んになり、今後益々データの交錯状態が増加し、更に大量のデータを高速に送信するために、パケット形式で、伝送されることが多くなっている。

【0003】このパケット形式によるヘッダ処理は、パケットの宛先のある出力インタフェースを求めると、該出力インタフェースに適するヘッダに変換するための情報を求めることにある。これらの作業は、ヘッダの各種フィールドなどとフォワーディングテーブルを用いて行うもので、非常に多数の処理ステップを必要とする。例えば、マルチレイヤスイッチ（これは物理レイヤの上位レイヤであるレイヤーL2とL3の処理を、1つの装置あるいはLSIチップにおいて行うスイッチである。）においては、L2のパケット（MACパケット）のヘッダからMAC（媒体アクセス制御：Media Access Control）のアドレスを取り出し、下記の4種類のMACフォーマットのどちらであるかを識別し、識別結果に応じて決まるヘッダ位置から、L3のパケット（IPパケット）を取り出し、IPパケットのヘッダからIPのアドレスを取り出す。

【0004】さらに、MACアドレスが所定のアドレスであるかどうかをテスト（ルータテスト）し、そうでないときには、L2の処理を行い、所定のアドレスであるときにはL3の処理を行う。前記識別結果によっては、L3のパケットを取り出す処理を行わず、L2処理を行

4

う。ルータテストについて、L2及びL3の処理では、フォワーディングテーブルを検索するという処理を行う。このフォワーディングテーブルは一般にかなり大きいため、この検索処理は非常に時間のかかるものとなる。

【0005】4種類のMACフォーマットとは、図14に示すように、Ethernet IIとも呼ばれるDIX規定とIEEE802.3規定という2つの標準形式と、それぞれに802.1Q規定とも呼ばれる仮想LANの識別子がついているかないかの2種類との積で4通りである。例えば、DIX規定と802.1Q規定との組合せでは、IP PDUとVLAN IDとがDIX規定の単独に対して余分に設定されている。また、LANエミュレーションの標準化の折、そのサービスは既存のイーサネット及びトークンリンクのLANのアプリケーションをエミュレートするもので、相互接続・運用の際にMACレベルでの接続が必須である。さらに、このMACパケットのペイロードとして、IPパケットがあり、さらにそのIPパケットのペイロードとして、TCPパケット、UDPパケットのいずれかがある。

【0006】このようにパケットヘッダは、それを順に解析して始めて、内容フォーマットが明らかになるものであるため、ヘッダ解析前のパケット受信時には、パケットヘッダがどこまでなのかは不明である。しかし、取り扱うプロトコルは定まっているため、最大のヘッダを想定して、この部分つまり、ヘッダを必ず含むようなパケットの一部分（ヘッダ可能性部分と呼ぶ）をパケットから取り出しておけば、不要な情報が含まれている可能性はあるが、必要な情報を取りこぼすことはない。さらに、検索結果を適当に加工して、パケットを宛先へ送出する準備を行う必要がある。この準備には、パケットを宛先行きの待ち行列に加える、パケットヘッダを書き換える、パケットをコピーするなどの処理がある。

【0007】以上のようなヘッダ処理を1つのCPUで順に行うことで、1つのパケットの処理が終了する。高速化を図るためには、この処理を高速に行う必要がある。

【0008】また、汎用のCPUを用いた場合には、論理回路で構成したハードウェアでは、1ステップ、つまり1クロックで実行できる機能を、2クロック以上必要とすることがある。

【0009】例えば、ハードウェア論理回路において、メモリAとメモリBの内容を2倍したものの加算を行い、メモリCに保存する（ $C = 2 \times A + 2 \times B$ ）には、1クロック目でメモリA、Bを同時に読み出してレジスタに入れ、2クロック目でそのレジスタを1ビットシフトしながら（2倍化）、加算し、その結果をメモリCに書き込むことができる（特殊演算と呼ぶ）。

【0010】一方、汎用CPUの場合には、加算やシフトといった最も基本的な演算（基本演算と呼ぶ）を用い

て、メモリ A をレジスタに読み出し（1 クロック目）、メモリ B をレジスタに読み出し（2 クロック目）、メモリ A のレジスタを 1 ビットシフトして（3 クロック目）、メモリ B のレジスタを 1 ビットシフトして（4 クロック目）、メモリ A のレジスタと B のレジスタを加算して（5 クロック目）、結果をメモリ C に書き込む（6 クロック目）ことになる。

【0011】この例で示すように、前者は 2 クロックで、後者は 6 クロックを要し、論理回路で構成したハードウェアは、汎用 CPU よりも高速に処理ができる。ただし、ハードウェア論理回路は、修正変更や将来の機能追加が困難である。そのため、ハードウェア論理回路は設計変更の柔軟性に欠ける。従来、ハードウェア論理回路におけるこの柔軟性を欠くという欠点を改善するためにシーケンサというプログラマブルな専用 CPU を用いる方法があり、信号処理などの分野で使用されていた。

【0012】シーケンサも汎用 CPU も、一般に、ROM/RAM に入れられたプログラムを持ち、プログラムカウンタと呼ばれるポインタが指し示すアドレスの内容（命令）が取り出され、命令が解釈（デコード）され、デコード信号が出され、デコード信号を受けた演算回路が動作するというものである。

【0013】しかしながら、シーケンサでは、演算回路が専用に作られており、例えば、レジスタ A とレジスタ B の内容をそれぞれ 2 倍して加算するという演算を、1 クロックで行なうような演算を組み込むことができる。従って、汎用 CPU よりも高速に演算が行われる。信号処理のシーケンサでは、X、Y を変数とし、A、B を媒介変数とする、 $Y = AX + B$ 等の一次関数の演算が演算回路として組まれている。しかしながら、パケットのヘッダを処理するためのシーケンサはなかった。

【0014】

【発明が解決しようとする課題】上記従来技術の第 1 の問題点は、ヘッダ処理において、順序よく行うべきことと、同時並行的に（独立に）行うことが可能ではあるが、汎用 CPU においては、すべて順に行われていたことにある。従って、高速化を図るには汎用 CPU の能力を上げる必要があるが、能力を上げることは、汎用 CPU そのものの性能アップのため、コスト高につながる。

【0015】また、汎用 CPU を多数用意し、パイプライン処理を行う場合には、適当なパイプライン数に設計しなければ、やはりコスト高になる。あるいはフォワーディングテーブルのような資源がボトルネックになり、汎用 CPU が能力を十分活かせなくなる恐れがある。

【0016】また、第 2 の問題点は、汎用 CPU のかわりにハードウェア論理回路を用いた場合には、設計変更の柔軟性に欠けるという問題がある。これに対し、シーケンサというプログラマブルな専用 CPU を用いる方法があり、信号処理などの分野で使用されていた。しかしながら、パケットのヘッダ処理を高速に行なうためのこ

のようなシーケンサは、いまだ公開されていない。

【0017】【発明の目的】本発明は、ボトルネックになるフォワーディングテーブルのような資源で制約される速度まで、パケット等のヘッダ処理の高速化を汎用 CPU を用いて低コストで行うことを目的とする。また、パケットヘッダ処理をプログラマブルなハードウェア論理回路、即ちシーケンス処理で実現することで、高速性を保ちながら、設計変更や将来の機能追加を低コストで実現することを目的とする。

10 【0018】

【課題を解決するための手段】上記目的を達成するために、本発明は、パケットから抽出されたヘッダを必ず含むようなパケットの一部分から、プロトコルヘッダの各フィールドのデータを抽出し、指定された記憶場所に入力する第 1 の手段と、該指定された記憶場所のデータを検索キーにして検索を行い、検索キーに一致するテーブルエントリの内容を出力する第 2 の手段と、該出力内容と該記憶場所のデータとから、パケットの送出先やヘッダ更新情報といった属性情報を作成する第 3 の手段とを

20

持つことを特徴とするヘッダ処理装置を提供するものである。

【0019】また本発明はパケットから抽出されたヘッダを含むパケットの一部分から、プロトコルヘッダの各フィールドのデータを抽出し、指定された記憶場所に入力する第 1 の手順と、該指定された記憶場所のデータを検索キーにして検索を行い、前記検索キーに一致するテーブルエントリの内容を出力する第 2 の手順と、該第 2 の手順の出力内容と前記記憶場所のデータとから、前記パケットの送出先とヘッダ更新情報の属性情報を作成する第 3 の手順と、を有することを特徴とするヘッダ処理方法を提供するものである。

30

【0020】【作用】パケットのヘッダ処理には、処理時間に注目すると主に以下の 3 つの処理が大半を占める。すなわち、第 1 の処理として、パケットのヘッダの構造を明らかにし、ヘッダの書くフィールドを抽出すること、第 2 の処理として、アドレスフィールドの情報、つまりアドレスからパケットの行き先を決定すること、第 3 の処理として、パケットの行き先へパケットを送出するのに必要なヘッダ変換を行うことである。

40

【0021】かかる処理を行う CPU の構成概念図を図 5 に示す。CPU による場合、CPU 10 内には加算器（+）と乗算器（×）とレジスタ R1、R2、R3 と命令デコーダとから成り、バスラインを介して CPU 10 の基本ソフトの OS 11 とプログラム 12 とから構成される。かかる構成の場合、プログラム 12 の内容の例によれば、A～D の数値と各レジスタとから、 $3A + 3B = D$ を得るのに 9 ステップを要しているが、コストパフォーマンスとしては申し分ない。

50

【0022】また、専用のシーケンスによる構成概念図を図 2 に示す。専用のシーケンスとしては、命令デコー

ダを有するシーケンサ20と、機能処理F1(21)、F2(22)と、シーケンサ20が動作する手順を格納するプログラム23とから構成される。かかる構成の場合、専用の機能処理手段を有しているので、図5と同様な結果を得るために、2ステップでよく、高速化のために優れている。

【0023】さらに、本発明により、パケットヘッダの一連の第1～第3の処理をパイプライン的に行うために、順に行うことが必要な前記3つの部分に分け、なおかつボトルネックになる資源の使用を最小限にとどめ、なおかつ該資源を使用するCPUが100%稼動するような分割とし、各部分では、それぞれのCPUを割り当てて独立に処理を行うことができる。これにより、最適なCPU数が選択でき、最も負荷がかかるCPUを100%あるいは100%近くまで活用することができる。

【0024】また、高速にパケットヘッダの処理を行うにはこれらを高速に行う必要がある。従来は、1つのCPUが上記3処理を順に行っていたが、ハードウェアで同様のことを行う場合、3処理を別々に行うことで、パイプライン処理が可能となる。

【0025】かかるパイプライン的な構成による処理の概念的な例を図3に示す。処理Aと処理B、処理Cとは同一の固定時間で処理が終わり、スタートしてから終了するまでも固定時間に設定されており、一連の処理が必要な場合、同一の固定時間で処理可能な処理Aと処理B、処理Cとがあり、CPU1で処理Aが終了してからCPU2で処理Bが開始されるとともにCPU1で次の処理Aが開始される、CPU2で処理Bが終了してからCPU3で処理Cが開始されるとともにCPU2で次の処理Bが開始されるような処理をパイプライン的な処理と称している。

【0026】また、処理A～Cが同一の固定時間で処理が終わらない場合に、プログラム処理する際には、種々な方式があり、その3例を図4に示して説明する。ここでは、CPUをシーケンサに置き換えて説明する。図4(a)は各処理の時間が異なる場合、処理プログラムA41が処理プログラムB42より短いとき、処理プログラムB42が終了するまでレジスタ43の処理プログラムB42に対応するフラグが立たないので、処理プログラムA41を処理するシーケンサ1は待ち受け状態とし、処理プログラムB42が終了してレジスタ43のフラグが立って、シーケンサ1は続く処理プログラムAを処理する。また、図4(b)では、処理プログラムA41が処理プログラムB42より短いとき、シーケンサ1は処理プログラムA41が終了してから所定の一定時間を待って、レジスタ43の処理プログラムB42に対応するフラグが立っているのかどうかを検出し、フラグが立った場合は次の処理プログラムA41を処理する。一方、フラグが立っていない場合は所定の一定時間を待ち、再度、レジスタ43の処理プログラムB42に対応

するフラグが立っているのかどうかを検出する。即ち、待ち受け状態に一定時間を設定しておくことで、シーケンサ1の負担を軽減している。さらに、図4(c)では、処理プログラムA41が処理プログラムB42より短いとき、所定の一定クロック数毎に、シーケンサ1がレジスタ43の処理プログラムB42に対応するフラグが立っているのかどうかを検出するもので、所定の一定クロック数の複数倍毎にフラグを検出することで、更にシーケンサの負担を軽減することができる。

10 【0027】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して詳細に説明する。

【0028】＜第1の実施形態＞本発明の第1の実施形態は、図1に示すように、第1～第3の手段となる3つの汎用CPUと各種レジスタから構成される。CPU(前処理)と名づけられた第1の汎用CPU101は、ヘッダレジスタ群(単にヘッダレジスタと称する)120の情報から、検索キーレジスタ群(単に検索キーレジスタと呼ぶ)121の内容を生成する。

20 【0029】CPU(検索(サーチ)/学習エンジン)と名づけられた第2の汎用CPU102は、検索キーレジスタ121の情報からフォワーディングテーブル109を検索し、検索結果により検索結果レジスタ群(単に検索結果レジスタと呼ぶ)122を生成する。

【0030】さらに、CPU(後処理)と名づけられた第3の汎用CPU103は、検索結果レジスタ122の情報からキューバッファ準備レジスタ群(単にキューバッファ準備レジスタと呼ぶ)123の内容を生成する。

30 【0031】ヘッダレジスタ120とキューバッファ準備レジスタ123は、受信インタフェース数、すなわち同時にパケットが到着する数Nだけ必要である。検索結果レジスタ122は、1つしかないフォワーディングテーブル109の検索に関係するので、N以下でよいが、任意の1つのレジスタのアクセスを2つの汎用CPUが同時に行わないという制約のもとで動作させるため、3つ必要である。

【0032】その3つの検索結果レジスタ122は、それぞれ、前処理汎用CPU101の書き込み、サーチ/学習エンジン102の書き込み、後処理汎用CPU103の読み取りに同時に使用される。この検索結果レジスタ122と対応させて制御させるため、検索キーレジスタ121も3つ用意する。ただし、検索キーレジスタ121においては、その2つは、それぞれ、前処理汎用CPU101の書き込み、検索エンジン102の読み取りに同時に使用され、1つは休止となる。

40 【0033】以下、3つの汎用CPUについて説明する。パケットあたりの処理において、検索/学習エンジン102がフォワーディングテーブル109の検索に時間がかかるため、もっともボトルネックとなる。また、この時間をできるだけ短縮するために、検索回数を最小

限にするための処理を前処理汎用CPU101と後処理汎用CPU103に任せる。

【0034】前処理と後処理のそれぞれの汎用CPU101、103の処理を説明する。前処理汎用CPU101は、ヘッダ解析を行い、MACパケットのヘッダフォーマットの決定および必要に応じて、MACパケットのペイロード、つまりIPパケットのヘッダ解析を行う。MACパケットのペイロードは必ずしもIPパケットではない。これは、MACアドレスがルータのMACアドレスであれば、該パケットはIPパケットとしてレイヤーL3を処理し、そうでなければIPパケットであるかどうかを調べる必要はなく、レイヤーL2の処理を行う。従来の技術では、MACアドレスがルータ宛てのMACアドレスかどうかは、フォワーディングテーブル109を検索していた。しかし、これは検索エンジン102の負荷になるので、この前処理汎用CPU101にて、その判定を行う。

【0035】また、判定でルータ宛先の解析が必要で、つまりL3の処理を行う必要がある場合には、IPパケットのヘッダの解析に入る。IPヘッダについては、宛先IPアドレスを抽出し、それを検索キーとして検索キーレジスタ121に入力する。また、受信ポート番号やVLAN-IDといった情報は、検索キーレジスタ121に書き込むとともに検索結果レジスタ122にも書き込む。これらの情報は、キューバッファ準備レジスタ123への書き込み時にも必要だからである。

【0036】後処理汎用CPU103は、検索結果レジスタ122の受信ポート番号と一致する番号のレジスタを指定するとともに、検索エンジン102により検索結果レジスタ122に入力された検索結果をキュー準備レジスタ123において指定されたフォーマットに変換する。例えば、宛先がマルチキャストである場合には、検索エンジン102は、フォワーディングテーブル109から、ビットマップを引き出してきて、それを検索結果レジスタ122に入力するので、それを後処理汎用CPU103が、そのビットマップを解析し、ビットのon/offをそのビットに対応する出力ポートへのパケット送出という命令列に変換する。

【0037】検索/学習エンジン102は、検索キーレジスタ121に入力された検索キーと、L2であるかL3であるかどうかの処理を行うべきかを示す情報を用いて、それに対応するフォワーディングテーブル109を検索する。検索により、キーと一致するものが発見された場合には、そのテーブル内で発見されたキーに結合されている情報を引き出す。該情報としては、出力ポート番号、出力時に変更すべきアドレスフィールド名と、変更用いるアドレス、などが含まれている。

【0038】以上のように3つの汎用CPUにより独立に処理を行うことで、一つの汎用CPUで処理を行う場合に問題になる高速化の限界を超えることができる。ま

た、汎用CPUを4つ以上使用する必要がなく、3つで十分に処理を分散することができる。

【0039】なお、上記実施形態では、処理の一部は前処理を経過して行う処理が必要であるが、重複して処理を行うことができるので、図5に示したCPUによりシーケンサ的にパケット処理の高速化が達成できる。

【0040】<第2の実施形態>本発明の第2の実施形態は、第1の実施形態における各汎用CPUの代わりにシーケンサを用いる構成である。これにより、汎用CPUを用いた場合よりもさらに高速化できる。図6にシーケンサによる処理の例を示している。シーケンサの動作及びシーケンサの構成は、図2に示したものであり、本実施形態による動作は、図1に示した第1の実施形態と同様なので、説明を省略する。なお、処理ステップが機能処理により極めて短時間で処理することが可能である。

【0041】<第3の実施形態>本発明の第3の実施形態は、第1あるいは第2の実施形態において、各CPU、又は各シーケンサがパイプラインとして動作し、以下のような構造を持つものである。図7は各シーケンサがパイプラインとして動作する場合を示す図である。

【0042】各シーケンサ101、102、103は、固定時間でその処理を終わるように固定時間(周期)を決めてある。システムを動作開始させるときに、同一のタイミングで信号が出るような周期そろえクロックを周期タイマ201から、シーケンサ101、102、103に同時に入力することで、周期の始まりを同一にする。これにより各シーケンサにおける処理は、次の手段のことを考慮せずに独立に動作することが可能となる。図7において、シーケンサに代えて汎用CPUを用いることも可能である。

【0043】これにより、汎用CPUあるいはシーケンサ制御において、汎用CPU間の連絡に必要な手続きが削除でき、或いは各シーケンサ間の連絡に必要な手続きが削除できるので、その分を本来の処理に向けることができ、さらなる高速化がはかれる。

【0044】<第4の実施形態>本発明の第4の実施形態は、図6の第2の実施形態において、図8に示すように、以下のような構造を持つものである。

【0045】各レジスタ群120~123は、それぞれ処理を行なうために必要なデータ、すなわち処理の入力データがあるかどうかを示す第1のレディレジスタをそれぞれ持つ。即ち、各レジスタ群120~123の各レジスタ120a~120c、121a~121c、122a~122c、123a~123cはそれぞれレディレジスタ301、302、303を持っている。さらに、各レジスタ群120~123は、処理の結果を出力するための記憶場所が使用できるかどうか、すなわち前パケットのヘッダ処理の結果が、次の手段で入力として使用済みかどうかを示す第2のレディレジスタをそれぞ

れ持つ。即ち、各レジスタ群120~123の各レジスタ120a~120c、121a~121c、122a~122c、123a~123cはそれぞれレディレジスタ401、402、403を持っている。

【0046】以下、上記構造の動作について説明する。

【0047】例えば、シーケンサ102は、処理の最初において、読み出し側レジスタ群（検索キーレジスタ群）121の第1のレジスタ121aの第1のレディレジスタ301と書き込み側レジスタ群（検索結果レジスタ群）122のレジスタ122aの第2のレディレジスタ401を常に読み出し、第1のレディレジスタ301が「入力データあり」の場合で、かつ、第2のレディレジスタ401が「書き込みOK」の場合にのみ、読み出し側レジスタ群121のレジスタ121aからデータを読み、データ処理して書き込み側レジスタ群122のレジスタ122aにデータを書き込む。その後、読み出し側レジスタ群121のレジスタ121aの第2のレディレジスタ401に「書き込みOK」をセットし、書き込み側レジスタ群122のレジスタ122aの第1のレディレジスタ301に「入力データあり」をセットする。さらに、読み出し側レジスタ群121のレジスタ121aの第1のレディレジスタ301に「入力データなし」をセットし、書き込み側レジスタ群122のレジスタ122aの第2のレディレジスタ401に「書き込み禁止」をセットする。

【0048】一方、読み出し側レジスタ群121のレジスタ121aの第1のレディレジスタ301が「入力データなし」の場合、あるいは書き込み側レジスタ群122のレジスタ122aの第2のレディレジスタ401が「書き込み禁止」の場合には、処理を開始せず、第1のレディレジスタ301あるいは第2のレディレジスタ401を読み出すだけのウエイト状態を続ける。

【0049】2つのシーケンサは少なくとも3つのレジスタを有し、これらの3つのレジスタは2つのシーケンサによって排他的に用いられる。例えば、シーケンサ101が検索キーレジスタ群121の1つのレジスタ121aのデータ書き込みを行っている場合、シーケンサ102は検索キーレジスタ群121の他の1つのレジスタ121bからデータの読み込みを行い、検索キーレジスタ群121の残りの1つのレジスタ121cは休止となる。

【0050】上記排他的動作について、各レジスタ群と各シーケンサの動作を示す図8を用いて、さらに説明する。

【0051】複数のシーケンサはあるレジスタを用いてデータを受け渡すのであるが、レジスタへの書き込みと読み出しと同時に起こると、データの妥当性がなくなる恐れがあるために、書き込みと読み出しを排他的に制御することが求められる。以下、排他的な動作について説明する。

【0052】例えば、検索キーレジスタ群121の動作に着目して説明すると、検索キーレジスタ群121の第1のレジスタ121aがシーケンサ101の書き込みに使用されているときには、検索キーレジスタ群121の第2のレジスタ121bは、シーケンサ102の読み出しに使用されており、検索キーレジスタ群121の第3のレジスタ121cはどこからもアクセスされない状態、またはシーケンサ101、102のどちらか一方の使用が可能な状態である。

10 【0053】シーケンサ101、102は、各々の処理を終了したら、次の処理のために別のレジスタを使用する。つまり、検索キーレジスタ群121の第2のレジスタ121bはシーケンサ101の書き込みに使用され、検索キーレジスタ群121の第3のレジスタ121cは、シーケンサ102の読み出しに使用されており、検索キーレジスタ群121の第1のレジスタ121aはどこからもアクセスされない状態、またはシーケンサ101、102のどちらか一方の使用が可能な状態となる。

20 【0054】このような動作を行うことで、各レジスタは読み込みと書き込みが排他的に行われるため、レジスタへの書き込みと読み出しが同時に起こることに起因するデータの妥当性の心配は解消される。

30 【0055】以上説明したように、本実施形態では、ハンドシェイクを用いた確実なパイプライン制御が行なえ、あるいは各手段で周期が異なることも可能であるし、周期の始まりを一致させる必要もなくなり、パイプライン制御の場合には、もっとも周期の長いものに全手段を合わせる必要があるのに比べて、より短時間で処理を終了することができる。なお、図8において、シーケンサに代えて汎用CPUを用いて第1の実施形態の構成に用いることも可能である。

40 【0056】＜第5の実施形態＞上記本発明の第4の実施形態においては、シーケンサ101~103の各シーケンサは、第1のレディレジスタが「入力データなし」、あるいは第2のレディレジスタが「書き込み禁止」の場合、常に読み出しを行うウエイト状態としていたが、本第5の実施形態では図4(b)を用いて説明した動作と同様に、決められたウエイト時間の間、読み出しを行わないウエイト状態になる構成とした。これにより、制御がさらに簡単になるため、そのぶん処理を高速化できる。

【0057】なお、本実施形態ではそれぞれあらかじめ決められた固定時間の始まりに現固定時間内にどの処理プログラムを実行するかを決定しても、固定時間の終わりに、次の固定時間内にどの処理プログラムを実行するかを、決定してもよい。

50 【0058】＜第6の実施形態＞本発明の第6の実施形態は、第5の実施形態において、シーケンサの処理時間が一定ではないが、ある決められた基準周期の整数倍となる場合である。

【0059】例えば、シーケンサ101とシーケンサ102との処理時間がともにある決められた基準周期の整数倍であるが、その処理時間が異なる場合に、シーケンサ101とシーケンサ102との処理時間の差も基準周期の整数倍となる。そのため、本実施形態では第5の実施形態における決められたウェイト時間を基準周期の整数倍とした。

【0060】例えば、図4(c)を用いて具体的に説明すると、例えば基準周期を100クロックとした場合、シーケンサ1は処理プログラムAの終了後、100クロック毎にレジスタを読み出して、処理プログラムB用のレジスタのフラグの状態を検出する。また、100クロック毎とせず、100クロックの整数倍の400クロック目にレジスタの状態を読み出し、フラグが立っていたならば、次の処理プログラムAの処理を開始する。これにより、処理を開始する必要のない時刻が決まるため、制御が簡単になるため、そのぶん処理を高速化できる。

【0061】＜第7の実施形態＞本発明の第7の実施形態は、図10に示すように、第2の実施形態において、第1のシーケンサ101が処理結果を蓄積するレジスタとして、キーレジスタ121と検索結果レジスタ122を両方使用し、第2のシーケンサ102が処理結果を蓄積するレジスタとして、検索結果レジスタ122を使用し、第3のシーケンサ103が処理結果を蓄積するレジスタとして、キュー準備レジスタ123を使用する構成である。つまり、検索結果レジスタ122は、第1のシーケンサ101と第2のシーケンサ102の両方から書き込みがあるレジスタである。ただし、検索結果レジスタ122は、該両方のシーケンサ101、102から同時にアクセスされることはない。これは、第1の実施形態でも述べたように、レジスタ122は、3個用意されており、3つのシーケンサそれぞれが、1つを読むあるいは書くことに使用しているためである。

【0062】第1のシーケンサ101は、第1のレジスタ121に検索に必要な情報を書き込み、同時に第2のレジスタ122に第3のシーケンサ103が必要とする情報(スルー情報)をパケットヘッダ可能部分から抽出して、書き込む。その他は、(CPUをシーケンサに書き換えた)第1の実施形態と同様である。

【0063】これにより、(CPUをシーケンサに書き換えた)第1の実施形態においては必要である第2のシーケンサ102がスルー情報をレジスタ122からレジスタ123へ移動させるという処理が必要ではなくなることで、制御が簡単になるため、そのぶん処理を高速化できる。

【0064】本実施形態で、シーケンサに代えて汎用CPUを用いる構成も可能である。また、シーケンサ101はサーチ/学習シーケンサ102が管理する検索結果レジスタ122に直接作用することを示したが、サーチ/学習シーケンサ102がキュー準備レジスタ123に

直接書き込むことも可能である。

【0065】＜第8の実施形態＞本発明の第8の実施形態は、図11、図12、図13に示すように、第2～7の実施形態における各シーケンサの行なう処理において、該処理は複数の命令から構成されており、図12に示すように、1つの命令の実行が加算やシフト算や、比較算のように汎用CPUにおいて用いられているような基本演算を用いて、あるいは1つの命令が、あるデータの加算した結果を比較しながら同時に別のデータをシフトするといったような該基本演算の並列あるいは順次的組み合わせで構成されている特殊演算を用いて実行される演算と命令で構成されている。これらの演算はどちらもハードウェアで構成されている。

【0066】以下、汎用CPUとシーケンサの処理クロック数の比較について説明する。

【0067】図11に示すように、汎用CPU600においては、命令セット604から取り出された命令は、レジスタ603の情報を用いて、第1の基本演算601あるいは第2の基本演算602を用いて演算結果を出し、再度レジスタ603に書き込む。

【0068】また、図12に示すように、本実施形態のシーケンサ700においては、命令セット704から取り出された命令は、レジスタ703の情報を用いて、特殊演算701あるいは基本演算602あるいは基本演算601を用いて演算結果を出し、再度レジスタ703に書き込む。

【0069】例えば、XとYという2つの情報があるレジスタにあり、この2つを用いて、 $Z = 2 \times X + 2 \times Y$ を求め、ZをレジスタZに書き込むという処理を行なう。基本演算としては、図12に示すように加算602及びシフト601を持っており、基本演算を用いて、命令を並べると以下になる。各命令は1クロックで実行する。レジスタXを1ビットシフトして(1クロック目)、レジスタYを1ビットシフトして(2クロック目)、レジスタXとレジスタYとを加算して(3クロック目)、結果をレジスタZに書き込む(4クロック目)。

【0070】一方、図13に示すように特殊演算701は、ビットシフト801とビットシフト802と加算803とを持ち、 $Z = 2 \times X + 2 \times Y$ を求め、ZをレジスタZ813に書き込むという処理を1クロックで行なう演算(演算Zと呼ぶ)である。従って、特殊演算を用いて命令を構成すると、演算Z(1クロック目)だけとなる。このように、あらかじめ必要な演算を特殊命令として、ハードウェアで実現しているため、処理の高速化が行なえる。一方、基本演算を組み合わせることで、特殊演算を実現できるようにしておけば、ハードウェアでは汎用CPUよりも同じ処理を高速に実現できるため高速化が行なえ、また命令の組み合わせの変更で演算の組み合わせの変更が可能になるため、柔軟性も得られ、設計

15

変更や新規処理の追加が高速性という利点を保ちつつ、低コストで実現できる。

【0071】上記各実施形態において、パケットのヘッダ処理装置及びヘッダ処理方法について説明したが、これらはMACヘッダやTCP/IPヘッダに限らず、他のヘッダ処理にも適用できるものである。また、汎用CPUやシーケンサの数を3個として説明したが、もっと多数の汎用CPU或いはシーケンサを用いることで更なる高速化を図ることができる。

【0072】また、各実施形態において、第1～第3の手段をCPU101～103又はシーケンサ101～103で構成した例を示したが、第1～第3の手段は1つのシーケンサと2つのCPU、2つのシーケンサと1つのCPU、独立した3つのシーケンサ、独立した3つのCPUで構成してもよく、また、共有される（第1～第3の手段のうち二つの手段が共有される）シーケンサと一つのシーケンサあるいは一つのCPU、共有されるCPUと一つのシーケンサあるいは一つのCPUで構成してもよい。図9は、図1において三つのCPUのうち一つのCPUを一つのシーケンサに置き換えた場合を示すシステムの概念ブロック図である。

【0073】

【発明の効果】本発明によれば、汎用CPUを用いてボトルネックになる資源で制約される速度までヘッダ処理が低コストで高速化できる。また、シーケンサを用いて、パケットヘッダ処理をプログラマブルなハードウェア論理回路で実現することで、高速性を保ちながら、設計変更や将来の機能追加を低コストで実現できる。さらに、複数の処理をパイプラインとして動作させて、ヘッダ処理、例えば次のルータの宛先の検索とパケットに宛先データの挿入等のルーティング処理等を高速に実行できる。

【図面の簡単な説明】

【図1】本発明の一実施形態のシステムの概念ブロック図である。

【図2】本発明のヘッダ処理のシステムの概念ブロック図である。

16

【図3】本発明のヘッダ処理のシステムの概念ブロック図である。

【図4】本発明のヘッダ処理のシステムの概念ブロック図である。

【図5】本発明のヘッダ処理のシステムの概念ブロック図である。

【図6】本発明の一実施形態のシステムの概念ブロック図である。

【図7】本発明の一実施形態のシステムの概念ブロック図である。

【図8】本発明の一実施形態のシステム処理の概念ブロック図である。

【図9】図5において一つのCPUを一つのシーケンサに置き換えた場合を示すシステムの概念ブロック図である。

【図10】本発明の一実施形態のシステムの概念ブロック図である。

【図11】本発明の一実施形態のシステムの概念ブロック図である。

【図12】本発明の一実施形態のシステムの概念ブロック図である。

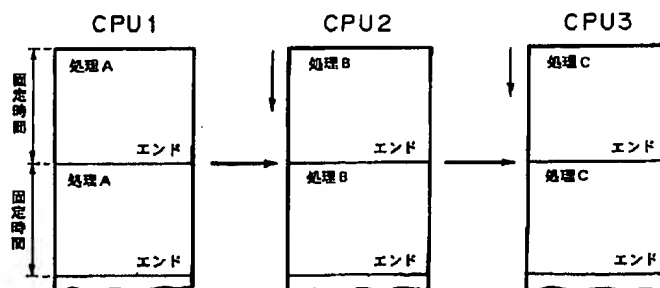
【図13】本発明の一実施形態のシステムの概念ブロック図である。

【図14】DIX規定とIEEE802.3規定の802.1Q規定とのフォーマットである。

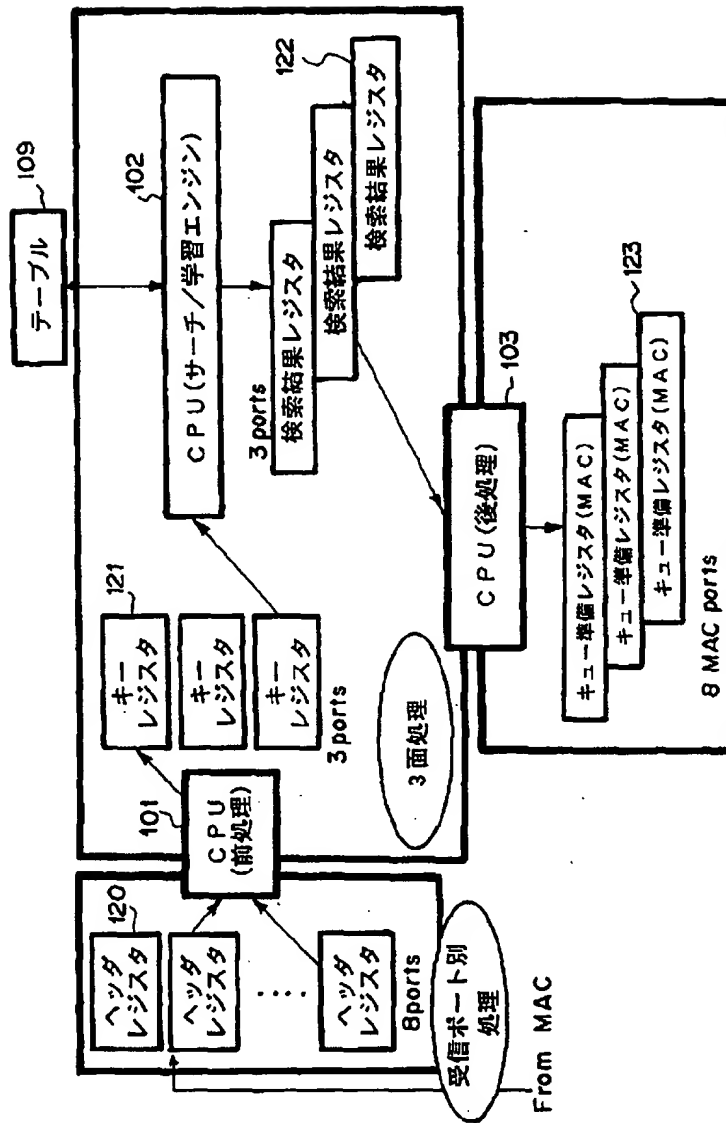
【符号の説明】

101 前処理シーケンサ
102 サーチ／学習エンジン
103 後処理シーケンサ
120 ヘッダレジスタ
121 検索キーレジスタ
122 検索結果レジスタ
123 キュー準備レジスタ
201 同期タイマ
301, 302, 303 レディレジスタ
401, 402, 403 レディレジスタ

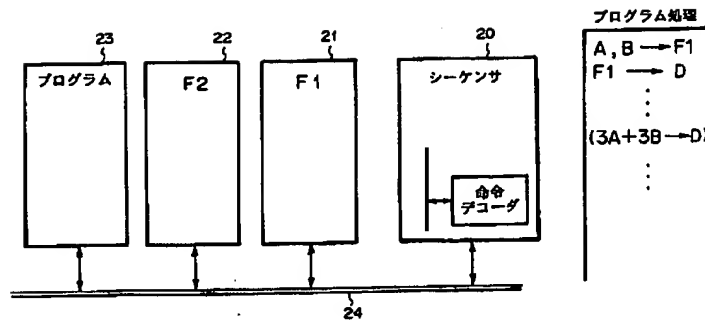
【図3】



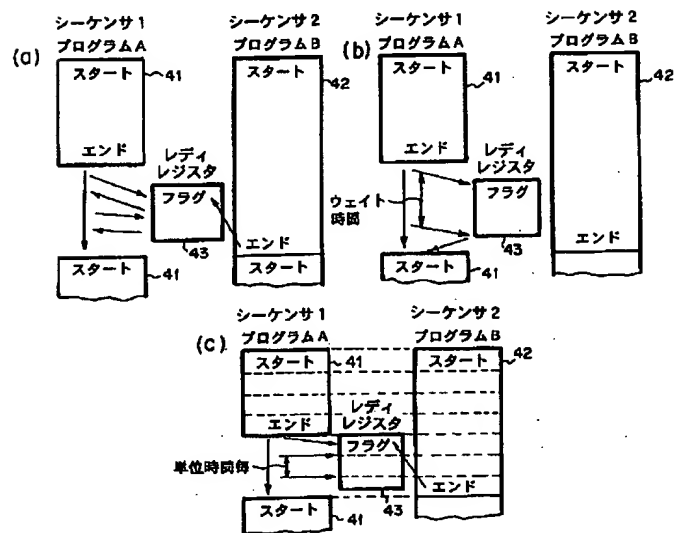
【図1】



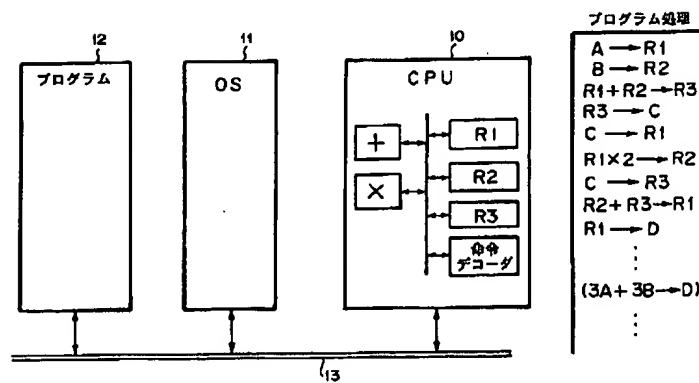
【図2】



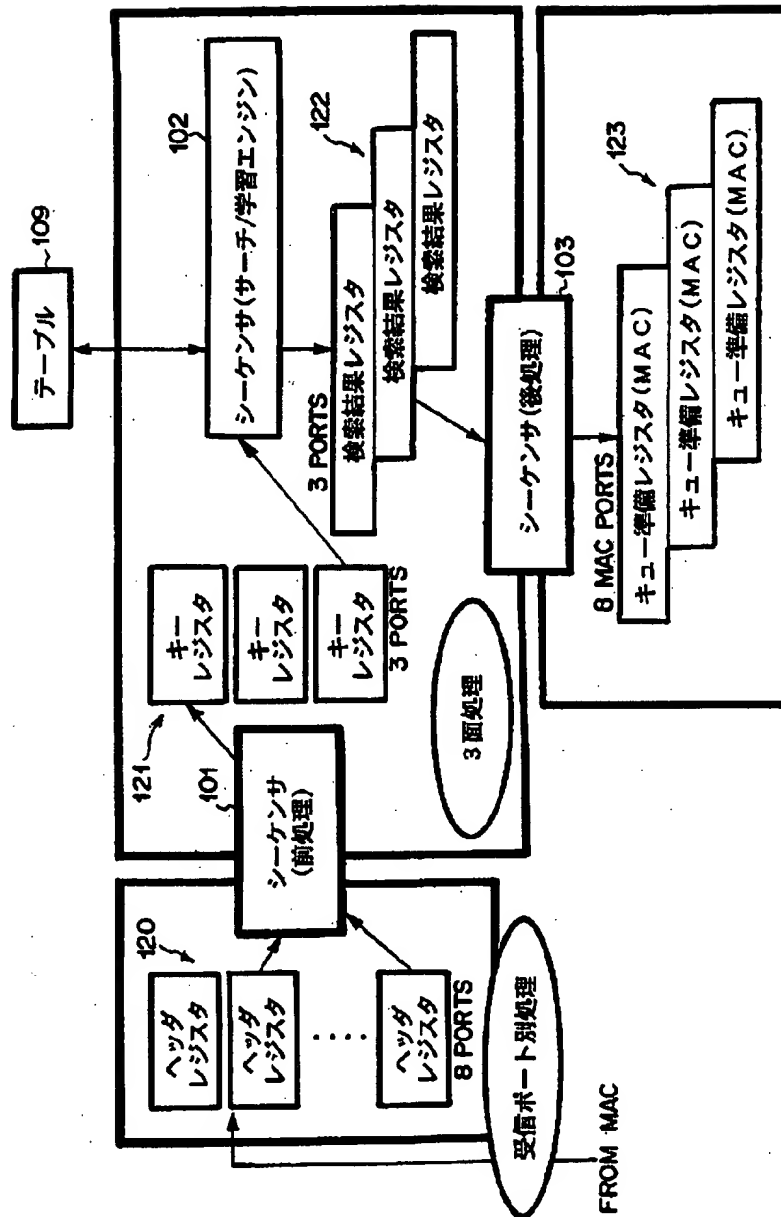
【図4】



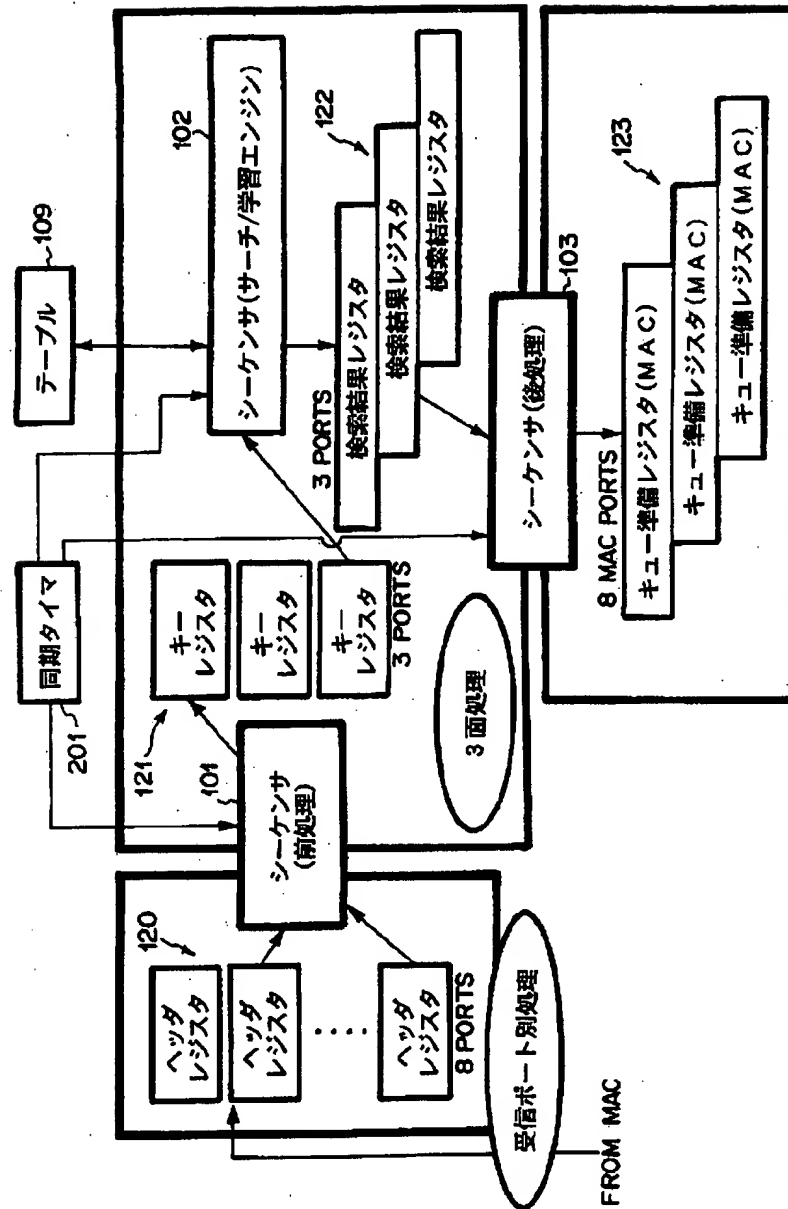
【図5】



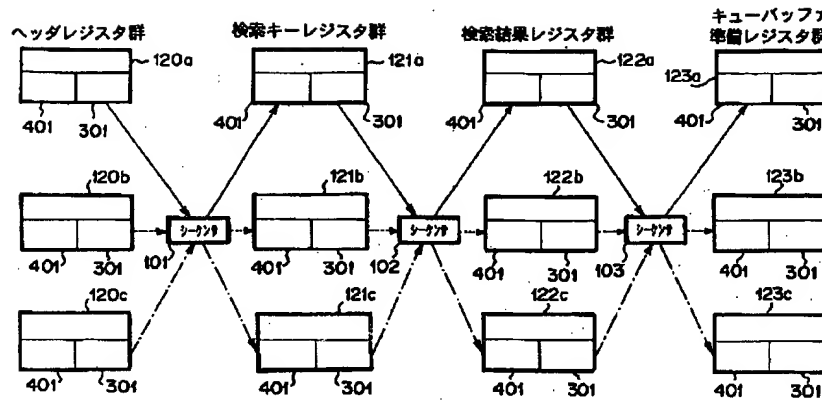
【図6】



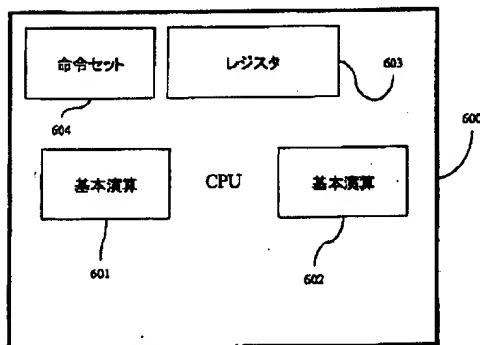
【図7】



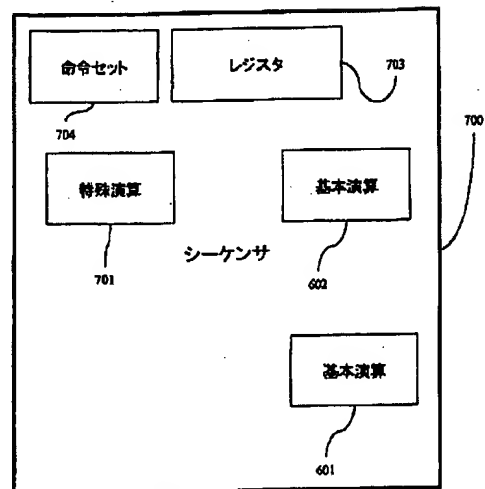
【図 8】



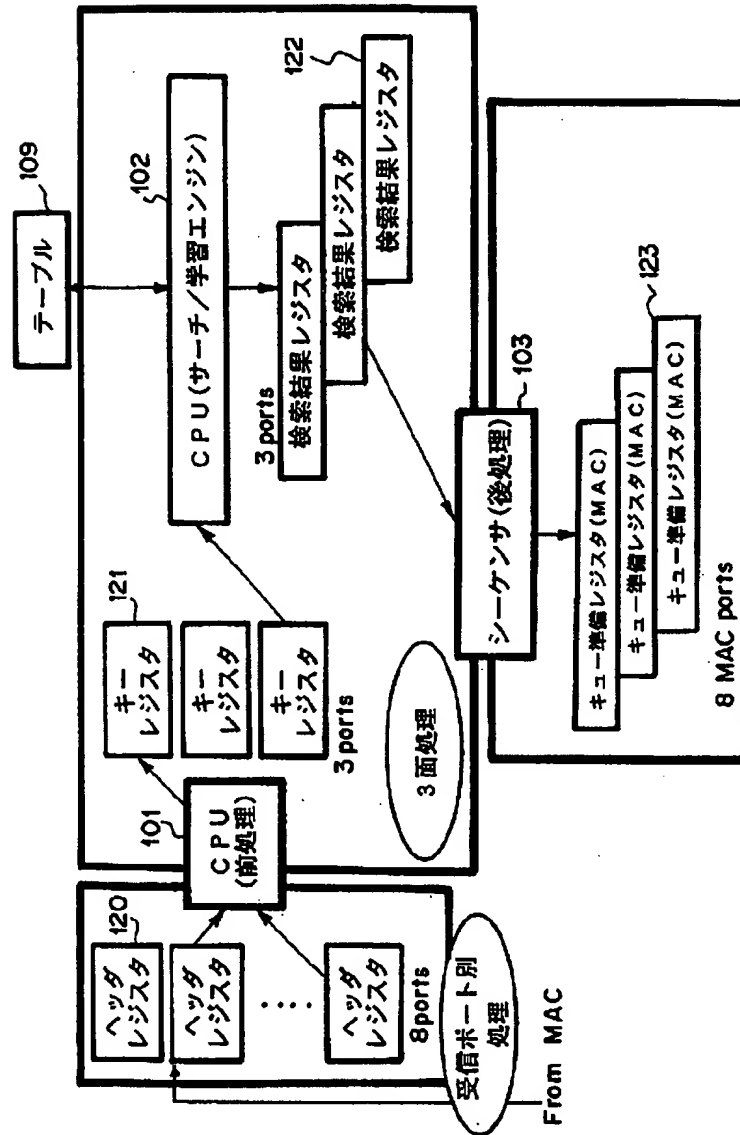
【図 11】



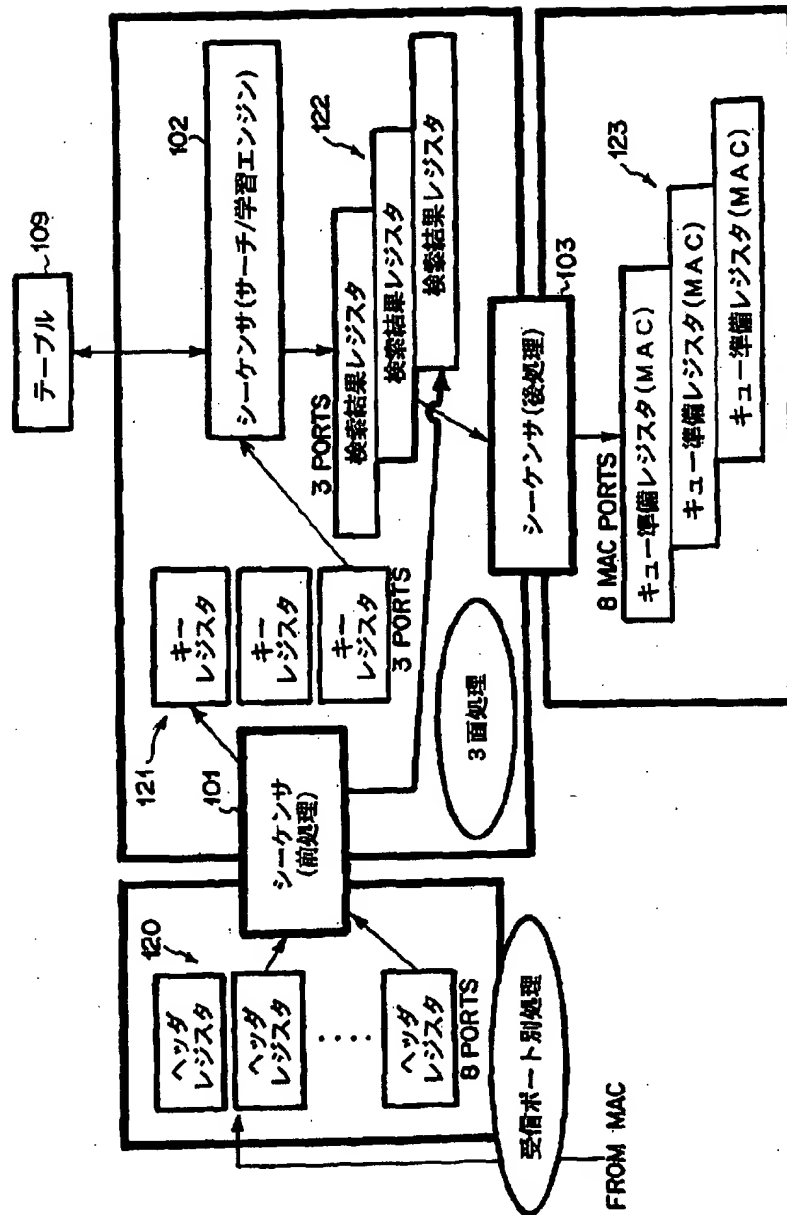
【図 12】



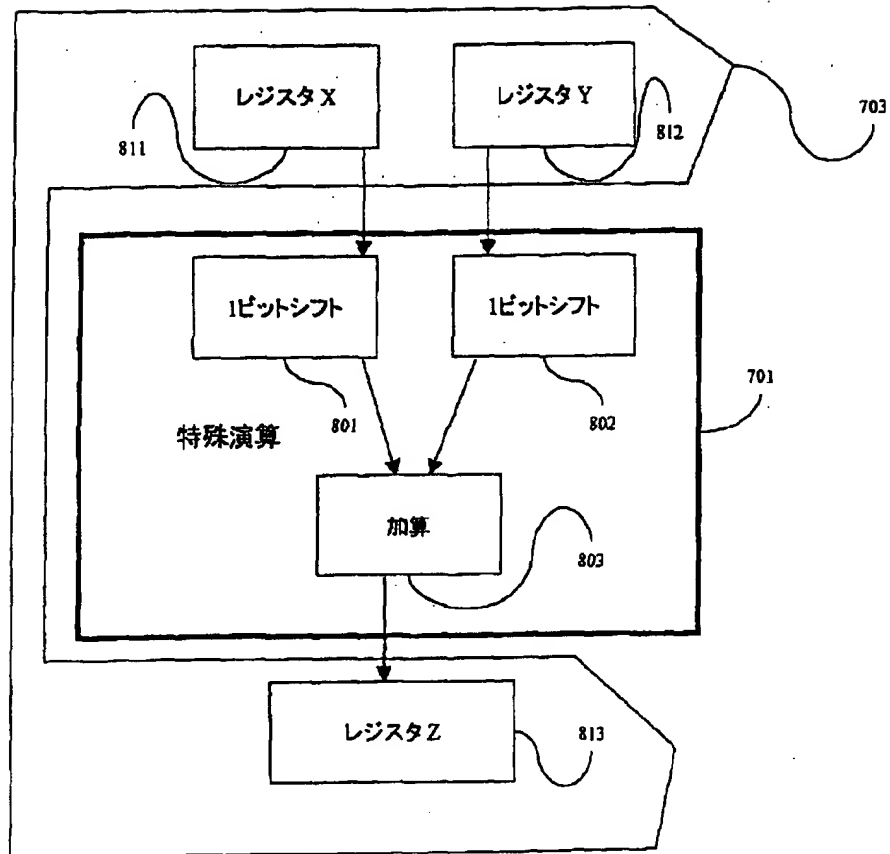
【図9】



【図10】



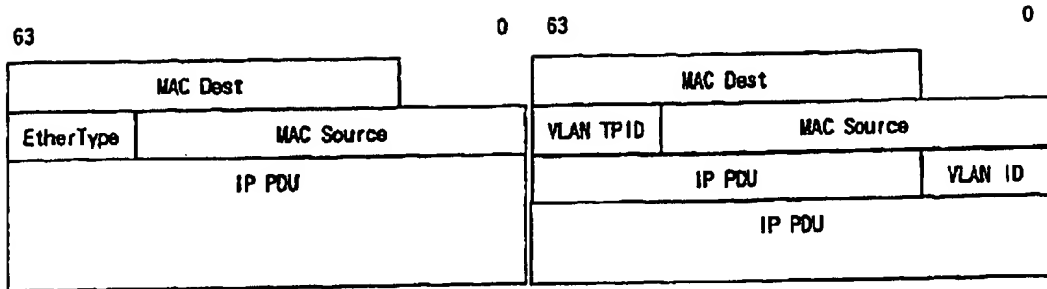
【図13】



【図 1 4】

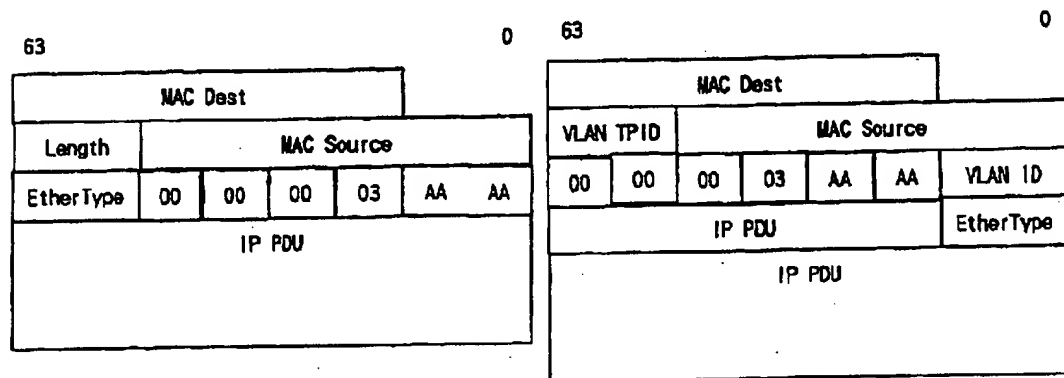
DIX Packet (non 802.1Q)

DIX Packet (with 802.1Q)



802.3 (no 802.1Q)

802.3 with 802.1Q



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.